Freeform Search

Database:

- US Pre-Grant Publication Full-Text Database
- US Patents Full-Text Database
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Term: `L6 and (cycles ADJ per ADJ instruction)`

Display: 50 **Documents in Display Format:** REV **Starting with Number** 1

Generate: ○ **Hit List** ● **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Sunday, January 07, 2007 Purge Queries Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L8	L6 and (cycles ADJ per ADJ instruction)	11	L8
L7	L6 and (cycles ADJ per ADJ instruction ADJ counter)	0	L7
L6	L5 and (optimizer or optimize or optimization)	159	L6
L5	L4 and threshold	1058	L5
L4	L3 OR l2	4411	L4
L3	cycle ADJ counter	3685	L3
L2	instruction ADJ counter	766	L2
L1	hardware ADJ performance	703	L1

END OF SEARCH HISTORY

Freeform Search

Database:
- US Pre-Grant Publication Full-Text Database
- **US Patents Full-Text Database**
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Term: L9 not 18

Display: 50 Documents in Display Format: REV Starting with Number 1

Generate: ○ Hit List ● Hit Count ○ Side by Side ○ Image

Search Clear Interrupt

Search History

DATE: Sunday, January 07, 2007 Purge Queries Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L10	L9 not 18	7	L10
L9	16 and (threshold).ab.	7	L9
L8	L6 and (cycles ADJ per ADJ instruction)	11	L8
L7	L6 and (cycles ADJ per ADJ instruction ADJ counter)	0	L7
L6	L5 and (optimizer or optimize or optimization)	159	L6
L5	L4 and threshold	1058	L5
L4	L3 OR 12	4411	L4
L3	cycle ADJ counter	3685	L3
L2	instruction ADJ counter	766	L2
L1	hardware ADJ performance	703	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
(717/151 |717/128).ccls. or (718/103 |718/108).ccls. or (711/128).ccls. or (714/47).ccls.	2727

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L11

Refine Search

Recall Text   Clear   Interrupt

Search History

DATE: Sunday, January 07, 2007 Purge Queries Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L11	717/151,128.ccls. or 718/103,108.ccls. or 711/128.ccls. or 714/47.ccls.	2727	L11
L10	L9 not L8	7	L10
L9	L6 and (threshold).ab.	7	L9
L8	L6 and (cycles ADJ per ADJ instruction)	11	L8
L7	L6 and (cycles ADJ per ADJ instruction ADJ counter)	0	L7
L6	L5 and (optimizer or optimize or optimization)	159	L6
L5	L4 and threshold	1058	L5
L4	L3 OR L2	4411	L4
L3	cycle ADJ counter	3685	L3
L2	instruction ADJ counter	766	L2
L1	hardware ADJ performance	703	L1

END OF SEARCH HISTORY

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

threshold optimizer CPU counter

SEARCH

Feedback Report a problem Satisfaction survey

Terms used threshold optimizer CPU counter

Found 10,575 of 193,448

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Memory systems: Reducing energy consumption of queries in memory-resident database systems

Jayaprakash Pisharath, Alok Choudhary, Mahmut Kandemir

September 2004 **Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems**

Publisher: ACM Press

Full text available: pdf(177.88 KB) Additional Information: full citation, abstract, references, index terms

The tremendous growth of system memories has increased the capacities and capabilities of memory-resident embedded databases, yet current embedded databases need to be tuned in order to take advantage of new memory technologies. In this paper, we study the implications of hosting memory resident databases, and propose hardware and software (query-driven) techniques to improve their performance and energy consumption. We exploit the structured organization of memories, which enables a selective m ...

Keywords: DRAM, database, energy, hardware schemes, layouts, mapping, power consumption, query optimization, query-directed energy management

2 DB-IR-1 (databases and information retrieval): indexing and query processing effiency: Energy management schemes for memory-resident database systems

Jayaprakash Pisharath, Alok Choudhary, Mahmut Kandemir

November 2004 **Proceedings of the thirteenth ACM international conference on Information and knowledge management CIKM '04**

Publisher: ACM Press

Full text available: pdf(251.47 KB) Additional Information: full citation, abstract, references, index terms

With the tremendous growth of system memories, memory-resident databases are increasingly becoming important in various domains. Newer memories provide a structured way of storing data in multiple chips, with each chip having a bank of memory modules. Current memory-resident databases are yet to take full advantage of the banked storage system, which offers a lot of room for performance and energy optimizations. In this paper, we identify the implications of a banked memory environment in sup ...

Keywords: DRAM, database, energy, hardware energy scheme, multiquery optimization, power consumption, query-directed energy management

3 The Performance of Runtime Data Cache Prefetching in a Dynamic Optimization System

Jiwei Lu, Howard Chen, Rao Fu, Wei-Chung Hsu, Bobbie Othmer, Pen-Chung Yew, Dong-Yuan Chen

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available: pdf(253.79 KB) Additional Information: full citation, abstract, citings, index terms

Traditional software controlled data cache prefetching isoften ineffective due to the lack of runtime cache miss andmiss address information. To overcome this limitation, weimplement runtime data cache prefetching in the dynamicoptimization system ADORE (ADaptive Object code RE-optimization).Its performance has been compared withstatic software prefetching on the SPEC2000 benchmarksuite. Runtime cache prefetching shows better performance.On an Itanium 2 based Linux workstation, it can increasepe ...

4 A Dynamic Compilation Framework for Controlling Microprocessor Energy and Performance

Qiang Wu, Margaret Martonosi, Douglas W. Clark, V. J. Reddi, Dan Connors, Youfeng Wu, Jin Lee, David Brooks

November 2005 **Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture MICRO 38**

Publisher: IEEE Computer Society

Full text available: pdf(517.60 KB)
Publisher Site Additional Information: full citation, abstract

Dynamic voltage and frequency scaling (DVFS) is an effective technique for controlling microprocessor energy and performance. Existing DVFS techniques are primarily based on hardware, OS timeinterrupts, or static-compiler techniques. However, substantially greater gains can be realized when control opportunities are also explored in a dynamic compilation environment. There are several advantages to deploying DVFS and managing energy/performance tradeoffs through the use of a dynamic compiler. Mo ...

5 GPGPU: general purpose computation on graphics hardware

David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: pdf(63.03 MB) Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

6 Profile-based optimizations: Dynamic trace selection using performance monitoring hardware sampling

Howard Chen, Wei-Chung Hsu, Jiwei Lu, Pen-Chung Yew, Dong-Yuan Chen

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '03**

Publisher: IEEE Computer Society

Full text available: pdf(1.88 MB) Additional Information: full citation, abstract, references, citings, index terms

Optimizing programs at run-time provides opportunities to apply aggressive optimizations to programs based on information that was not available at compile time. At run time, programs can be adapted to better exploit architectural features, optimize the use of dynamic libraries, and simplify code based on run-time constants.Our profiling system provides a framework for collecting information required for performing run-time optimization. We sample the performance hardware registers available on ...

7 Ispike: A Post-link Optimizer for the Intel®Itanium®Architecture

Chi-Keung Luk, Robert Muth, Harish Patil, Robert Cohn, Geoff Lowney
March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '04**
Publisher: IEEE Computer Society
Full text available: pdf(286.96 KB) Additional Information: full citation, abstract, citings, index terms

Ispike is post-link optimizer developed for theIntel®Itanium Processor Family (IPF) processors.TheIPF architecture poses both opportunities and challenges topost-link optimizations.IPF offers a rich set of performancecounters to collect detailed profile information at a low cost,which is essential to post-link optimization being practical.At the same time, the prediction and bundling features onIPF make post-link code transformation more challengingthan on other architectures.In Ispike, we have ...

8 Method-level phase behavior in java workloads

Andy Georges, Dries Buytaert, Lieven Eeckhout, Koen De Bosschere
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10
Publisher: ACM Press
Full text available: pdf(695.63 KB) Additional Information: full citation, abstract, references, index terms

Java workloads are becoming more and more prominent on various computing devices. Understanding the behavior of a Java workload which includes the interaction between the application and the virtual machine (VM), is thus of primary importance during performance analysis and optimization. Moreover, as contemporary software projects are increasing in complexity, automatic performance analysis techniques are indispensable. This paper proposes an off-line method-level phase analysis approach for ...

9 Query evaluation techniques for large databases

Goetz Graefe
June 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 2
Publisher: ACM Press
Full text available: pdf(9.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

10

Cache investment: integrating query optimization and distributed data placement
Donald Kossmann, Michael J. Franklin, Gerhard Drasch, Wig Ag
December 2000 **ACM Transactions on Database Systems (TODS)**, Volume 25 Issue 4
Publisher: ACM Press
Full text available: pdf(210.67 KB) Additional Information: full citation, abstract, references, citings, index terms

Emerging distributed query-processing systems support flexible execution strategies in which each query can be run using a combination of data shipping and query shipping. As in any distributed environment, these systems can obtain tremendous performance and availability benefits by employing dynamic data caching. When flexible execution and dynamic caching are combined, however, a circular dependency arises: Caching occurs as a by-product of query operator placement, but query operator pl ...

Keywords: cache investment, caching, client-server database systems, data shipping, dynamic data placement, query optimization, query shipping

11 Research sessions: query optimization: Robust query processing through progressive optimization
Volker Markl, Vijayshankar Raman, David Simmen, Guy Lohman, Hamid Pirahesh, Miso Cilimdzic
June 2004 **Proceedings of the 2004 ACM SIGMOD international conference on Management of data**
Publisher: ACM Press
Full text available: pdf(331.15 KB) Additional Information: full citation, abstract, references

Virtually every commercial query optimizer chooses the best plan for a query using a cost model that relies heavily on accurate cardinality estimation. Cardinality estimation errors can occur due to the use of inaccurate statistics, invalid assumptions about attribute independence, parameter markers, and so on. Cardinality estimation errors may cause the optimizer to choose a sub-optimal plan. We present an approach to query processing that is extremely robust because it is able to detect and re ...

12 Real-time shading
Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(7.39 MB) Additional Information: full citation, abstract

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders of thousands to tens of thousands of instructions. This course has been redesigned to address today's real-time shading capabili ...

13 Design and evaluation of dynamic optimizations for a Java just-in-time compiler
Toshio Suganuma, Toshiaki Yasue, Motohiro Kawahito, Hideaki Komatsu, Toshio Nakatani
July 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 4
Publisher: ACM Press
Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, index terms

The high performance implementation of Java Virtual Machines (JVM) and Just-In-Time (JIT) compilers is directed toward employing a dynamic compilation system on the basis of online runtime profile information. The trade-off between the compilation overhead and performance benefit is a crucial issue for such a system. This article describes the design and implementation of a dynamic optimization framework in a production-level Java JIT compiler, together with two techniques for profile-directed o ...

Keywords: JIT compiler, Recompilation, adaptive optimization, code specialization, dynamic compilation, profile-directed method inlining

14 Managing bounded code caches in dynamic binary optimization systems
Kim Hazelwood, Michael D. Smith
September 2006 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 3 Issue 3
Publisher: ACM Press
Full text available: pdf(666.72 KB) Additional Information: full citation, abstract, references, index terms

Dynamic binary optimizers store altered copies of original program instructions in software-managed code caches in order to maximize reuse of transformed code. Code caches store code blocks that may vary in size, reference other code blocks, and carry a high replacement overhead. These unique constraints reduce the effectiveness of conventional cache management policies. Our work directly addresses these unique constraints and presents several contributions to the code-cache management problem. ...

Keywords: Dynamic optimization, code caches, dynamic translation, just-in-time compilation

15 Dynamo: a transparent dynamic optimization system
Vasanth Bala, Evelyn Duesterwald, Sanjeev Banerjia
May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation PLDI '00**, Volume 35 Issue 5
Publisher: ACM Press
Full text available: pdf(156.03 KB) Additional Information: full citation, abstract, references, citings, index terms

We describe the design and implementation of Dynamo, a software dynamic optimization system that is capable of transparently improving the performance of a native instruction stream as it executes on the processor. The input native instruction stream to Dynamo can be dynamically generated (by a JIT for example), or it can come from the execution of a statically compiled native binary. This paper evaluates the Dynamo system in the latter, more challenging situation, in order to emphasize the....

16 VHC: Quickly Building an Optimizer for Complex Embedded Architectures
Michael Dupré, Nathalie Drach, Olivier Temam
March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '04**
Publisher: IEEE Computer Society
Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, index terms

To meet the high demand for powerful embedded processors,VLIW architectures are increasingly complex (e.g.,multiple clusters), and moreover, they now run increasinglysophisticated control-intensive applications. As a result, developingarchitecture-specific compiler optimizations is becomingboth increasingly critical and complex, while time-to-market constraints remain very tight.In this article, we present a novel program optimizationapproach, called the Virtual Hardware Compiler (VHC),that can ...

17 A region-based compilation technique for dynamic compilers
Toshio Suganuma, Toshiaki Yasue, Toshio Nakatani
January 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 28 Issue 1
Publisher: ACM Press
Full text available: pdf(977.63 KB) Additional Information: full citation, abstract, references, index terms

Method inlining and data flow analysis are two major optimization components for effective program transformations, but they often suffer from the existence of rarely or never executed code contained in the target method. One major problem lies in the assumption that the compilation unit is partitioned at method boundaries. This article describes the design and implementation of a region-based compilation technique in our dynamic optimization framework, in which the compiled regions are selected ...

Keywords: JIT compiler, Region-based compilation, dynamic compilation, on-stack replacement, partial inlining

18 Speeding up construction of PMR quadtree-based spatial indexes
Gisli R. Hjaltason, Hanan Samet
October 2002 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 11 Issue 2
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(355.72 KB) Additional Information: full citation, abstract, citings, index terms

Spatial indexes, such as those based on the quadtree, are important in spatial databases for efficient execution of queries involving spatial constraints, especially when the queries involve spatial joins. In this paper we present a number of techniques for speeding up the construction of quadtree-based spatial indexes, specifically the PMR quadtree, which can index arbitrary spatial data. We assume a quadtree implementation using the "linear quadtree", a disk-resident representation ...

Keywords: Bulk-loading, I/O, Spatial indexing

19 Track 7: compilers and operating systems: Dynamic run-time architecture techniques for enabling continuous optimization
Tipp Moseley, Alex Shye, Vijay Janapa Reddi, Matthew Iyer, Dan Fay, David Hodgdon, Joshua L. Kihm, Alex Settle, Dirk Grunwald, Daniel A. Connors
May 2005 **Proceedings of the 2nd conference on Computing frontiers**
Publisher: ACM Press
Full text available: pdf(331.25 KB) Additional Information: full citation, abstract, references, index terms

Future computer systems will integrate tens of multithreaded processor cores on a single chip die, resulting in hundreds of concurrent program threads sharing system resources. These designs will be the cornerstone of improving throughput in high-performance computing and server environments. However, to date, appropriate systems software (operating system, run-time system, and compiler) technologies for these emerging machines have not been adequately explored. Future processors will require so ...

Keywords: multithreading, performance counters, profiling, scheduling

20 Mixed mode execution with context threading
Mathew Zaleski, Marc Berndl, Angela Demke Brown
October 2005 **Proceedings of the 2005 conference of the Centre for Advanced Studies**

on Collaborative research CASCON '05

Publisher: IBM Press

Full text available: pdf(162.98 KB) Additional Information: full citation, abstract, references, index terms

Interpreters are widely used to implement portable language runtime environments. Programs written in these languages may benefit from performance beyond that obtainable by optimizing interpretation alone. A modern high-performance mixed-mode virtual machine (VM) includes amethod-based Just In Time (JIT) compiler. A method-based JIT, however, requires the up-front development of a complex compilation infrastructure before any performance benefits are realized.Ideally, the architecture for a mixe ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next